

Architecture Assessment and Product Quality

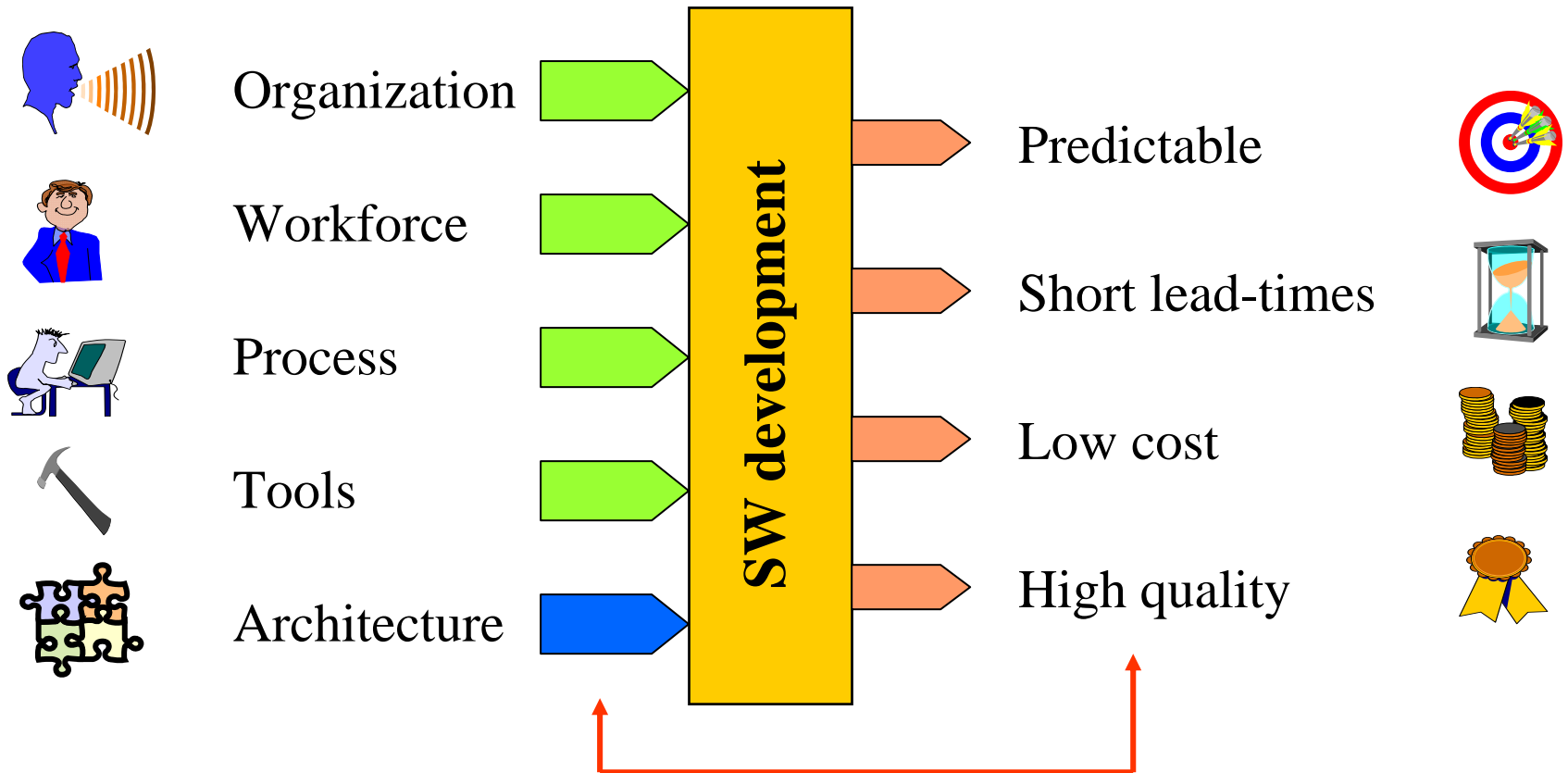
SPIDER Plenary Session 25/05/05

Herman Postema

Herman.Postema@LogicaCMG.com

Capabilities

Performance



- Introduction
- Architecture Assessment
- Case Philips RCS
- Conclusions

Product quality

- Satisfy customer requirements / expectations
- Can be expressed in quality properties, also called
 - Quality factors, quality characteristics
 - Non-functional properties
 - “ilities”
- Modelled and defined in standards like
 - ISO / IEC 9126 : *Software Engineering – Product Quality*
 - IEEE 1061 : *Standard for a Software Quality Metrics Methodology*
 - QUINT (2) : *Model voor kwaliteitsattributen*



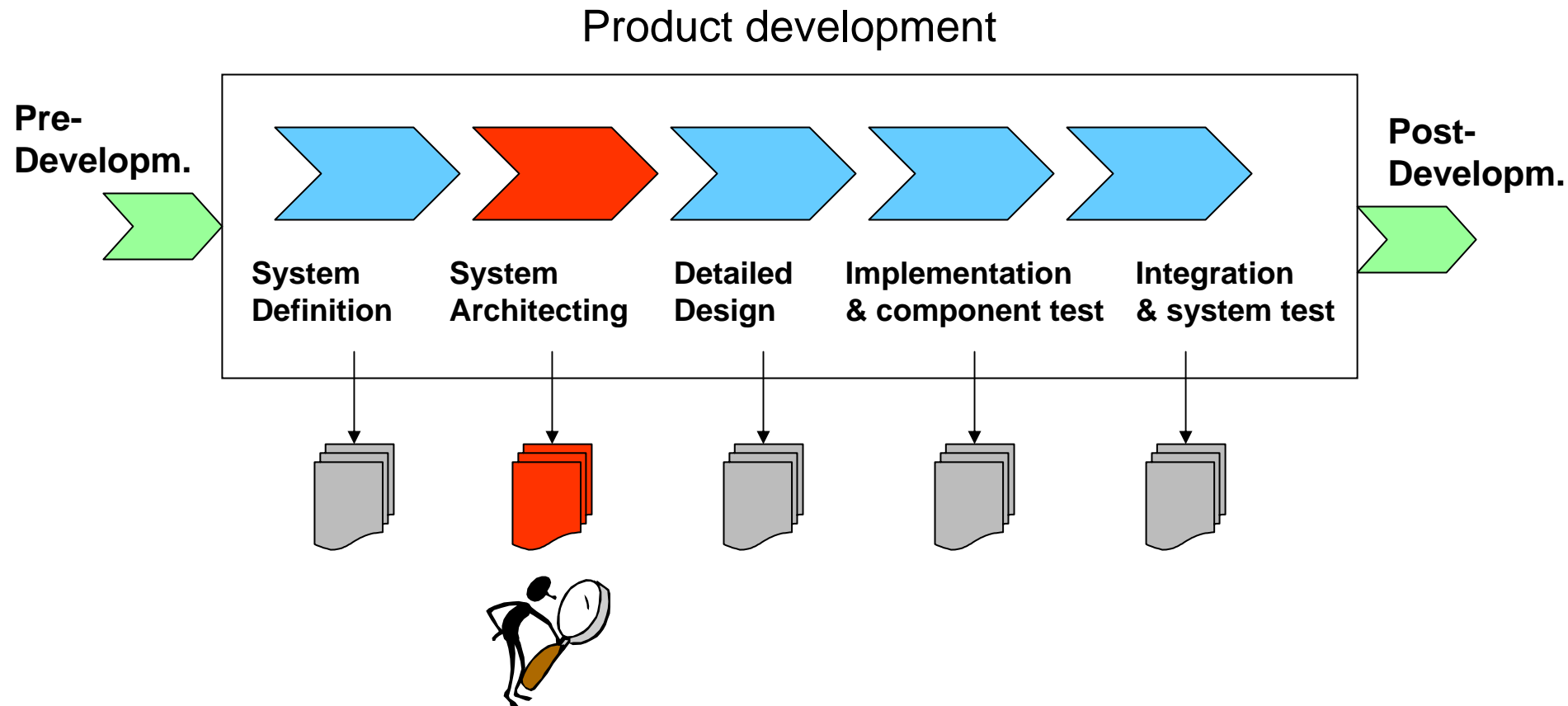
Examples of quality characteristics

- Reliability
- Availability
- Maintainability
- Extend ability
- Portability
- Outsource ability
- Testability
- Reusability
- Serviceability
- Time performance
- Resource efficiency (memory)
- Installability
- Upgradeability
- Scalability
- Ease of Creation
- Buy-in
- Conformance
- Security
- Compatibility
- Safety
- Usability
- Stability

Architecture

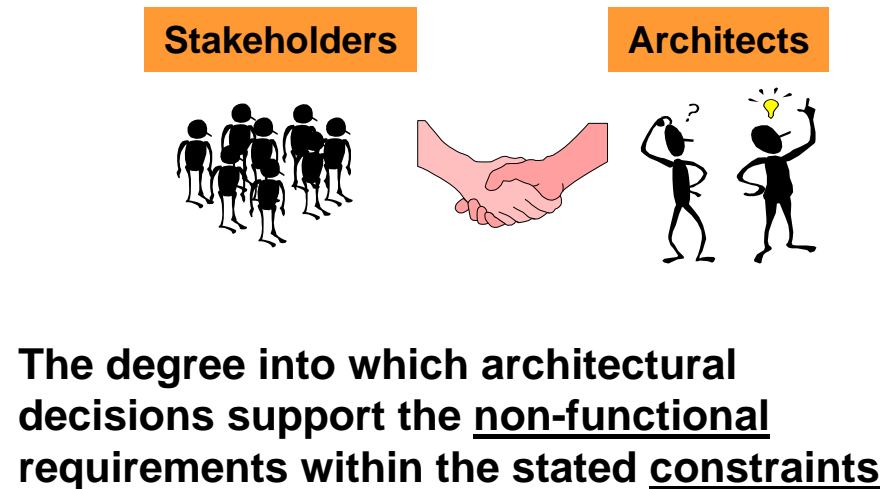
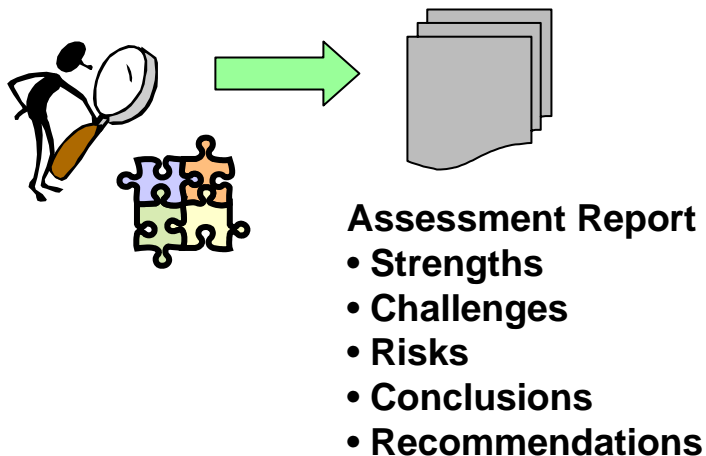
- Related to (IEEE 1471)
 - The organization of a system into components
 - The relationships between these components and to the environment
 - The principles guiding its design and evolution
- The technical solution for a problem or need
 - The “design” of the system (the “how”)
 - Between “specification” and “implementation”
- Justifies its existence primarily to the non-functional (or quality) requirements of the product

Architecture as the first “predictor” of product quality



- Goal
- Approach
- Benefits
- Applications
- Experiences & Lessons Learned
- Findings

To determine the quality of a system / SW architecture and provide recommendations for improvement



Approach (1)

Phasing

Preparation

Assessment Plan
Readiness Check
Participant briefing

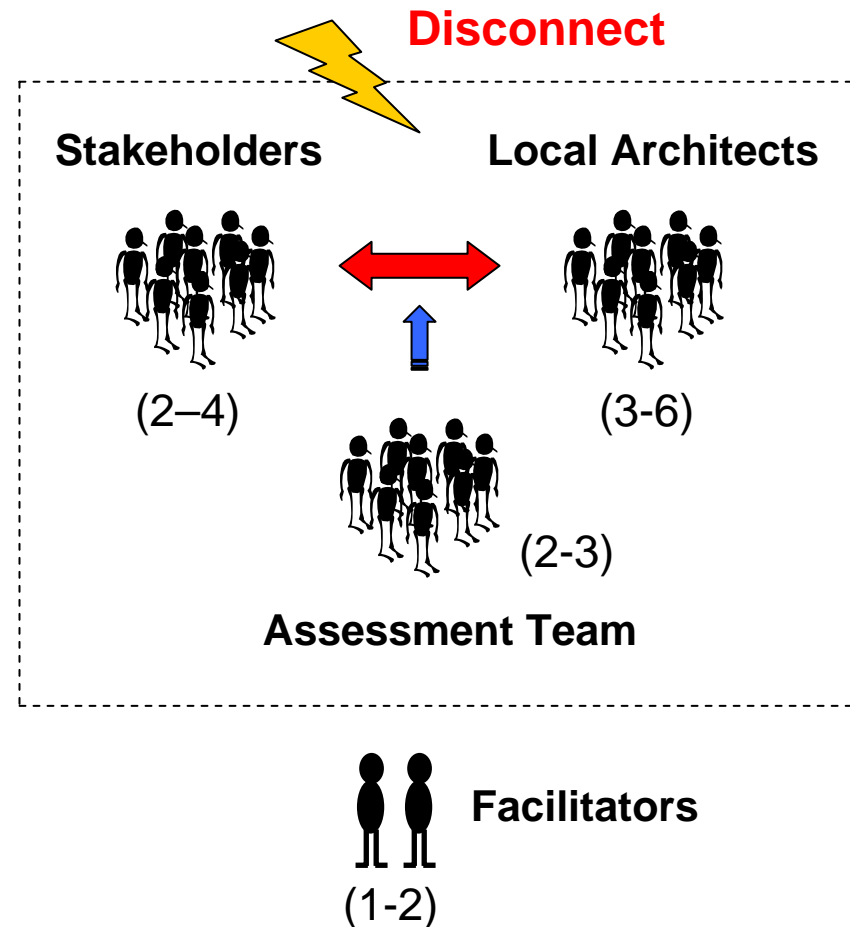
Review

Problem statement
Context building
Issue exploration
Finalization
Wrap-up

Reporting

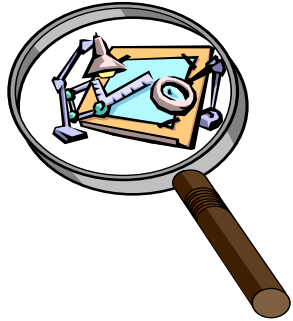
Report preparation
Feedback
Evaluation

Organization



Approach (2)

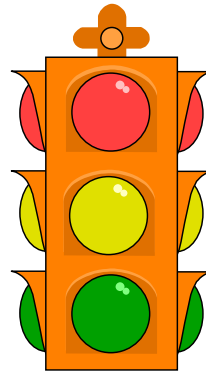
N.F. Requirement	Priority	Keuze															
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16
		OS	Almelo	command	printable	plug&play	compatib	RFFI	structure	modem	Modula-2	logging	ROM-tod	Simulatie	DOS	Version	- method
Ease of Creation	Must	++	++	++	+	+	-	-	+	--	++	++	+	++	-/+	+	--
Independence	Not Important																
Concurrent Dev.	?	++	++	.	.	+	.	.	+	.	++	--
Outsourceability	Not Exclude	-	-	+/-	.	.	-	-	+/-	.	-	.	-	++	.	.	-
Buy-in	Must	--	-	.	-
Manufacturability	Not Important																
Interoperability	Not Important																
Security	Must	.	+	.	-	-
Configurability	Must	++	+	+
Compatibility	Must	++	++	++	+/-	.	+	-
Correctness	Must	+	.	+	+	.	.	--
Availability	Must	+/-	+	-
Fault tolerance	Must	+	+	+
Recoverability	Not Important																
Safety	Must	++	-
Time behaviour	Must	++	-	.	-	.	.	.	+
Resource utilization	Not Important																
Correctability	Must	.	.	+	++	+	.	.	+	.	++	+	+	++	.	.	--
Expandability	Must	+	.	.	.	+	+	+	.	.	+	+	-
Stability	Must	.	+	+	+	.	+	--
Testability	Must	.	.	+	+	.	.	.	+	++	.	.	-
Future Proof	Important	-	-	-	.	-	.	-	.	.
Adaptability	Important	+	-	.	.	+	+	+
Installability	Important	.	-	.	-	+	.	.	+
Reusability	Must	++	++	+	.	.	+	-



Architecture findings



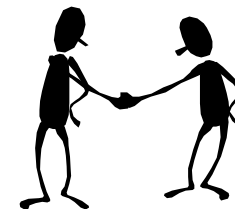
Process findings



Sense of urgency



Awareness & learning



Mutual understanding & team building

- PMS/XRD *‘Evaluate the quality of the Fusion platform architecture in general’*
- PA X-ray *‘Identify the architectural concepts to be reused in our next generation of products’*
- CSI *‘Evaluate the highest priority ASRs of our next platform release’*
- RCS *‘Identify architectural risks that may danger successful realization of our next project release’*
- PCE/TV *‘Determine how well MGR exploits the commonality in our product roadmaps’*
- Bosch *‘Evaluate the findings of our internal ATAM assessment’*
- PMS/MR *‘Evaluate adequacy and implementation of two key technology choices (Wintel, MIP)’*
- Bosch *Identify architectural risks that may threaten the successful realization of the Plato project*

- Need to focus
 - Content, stakeholders, issues, choices
- No fixed approach
 - Tailor to context, needs, constraints
 - Frequent evaluation and adjustment
- Thorough preparation
 - Verify assessment “readiness”
 - Manage expectations
 - Carefully select participants (de-politicize)
 - Motivate participants



Concepts

Terminology

Basic steps

Phases

Variations

Best practices

Tools



- Added value is generally perceived high
 - Assessment provides transparency to management
 - Well motivated insight into architectural risks
 - Findings (partly recognized) are made explicit
 - Involvement of external architects
 - Findings are supported by the local architects
- Opportunities for improvement
 - Selection of participants
 - More extensive preparation by external architects
 - Need for more detailed investigations



Architecture findings

- Quality of (f+nf) specifications
- Balance short – long term
- Balance SW – HW architecture
- Quality of decomposition
- Execution architecture
- Preparing for innovations
- Design erosion
- Pragmatics (“too ...”)
- Quality of interfaces
- Quality of documentation



Process findings

- Architecture awareness
- Skills of architects
- Architectural leadership (roadmaps)
- Definition of architecture process
- Architects \Leftrightarrow stakeholders
- Managing non-functional requirements
- Architectural choices and trade-offs
- Architecture evaluation
- Guarding design & implementation
- Architecture documentation

- Context
- Purpose
- Scope
- Organization
- Workshop Agenda
- Aspects
- Findings

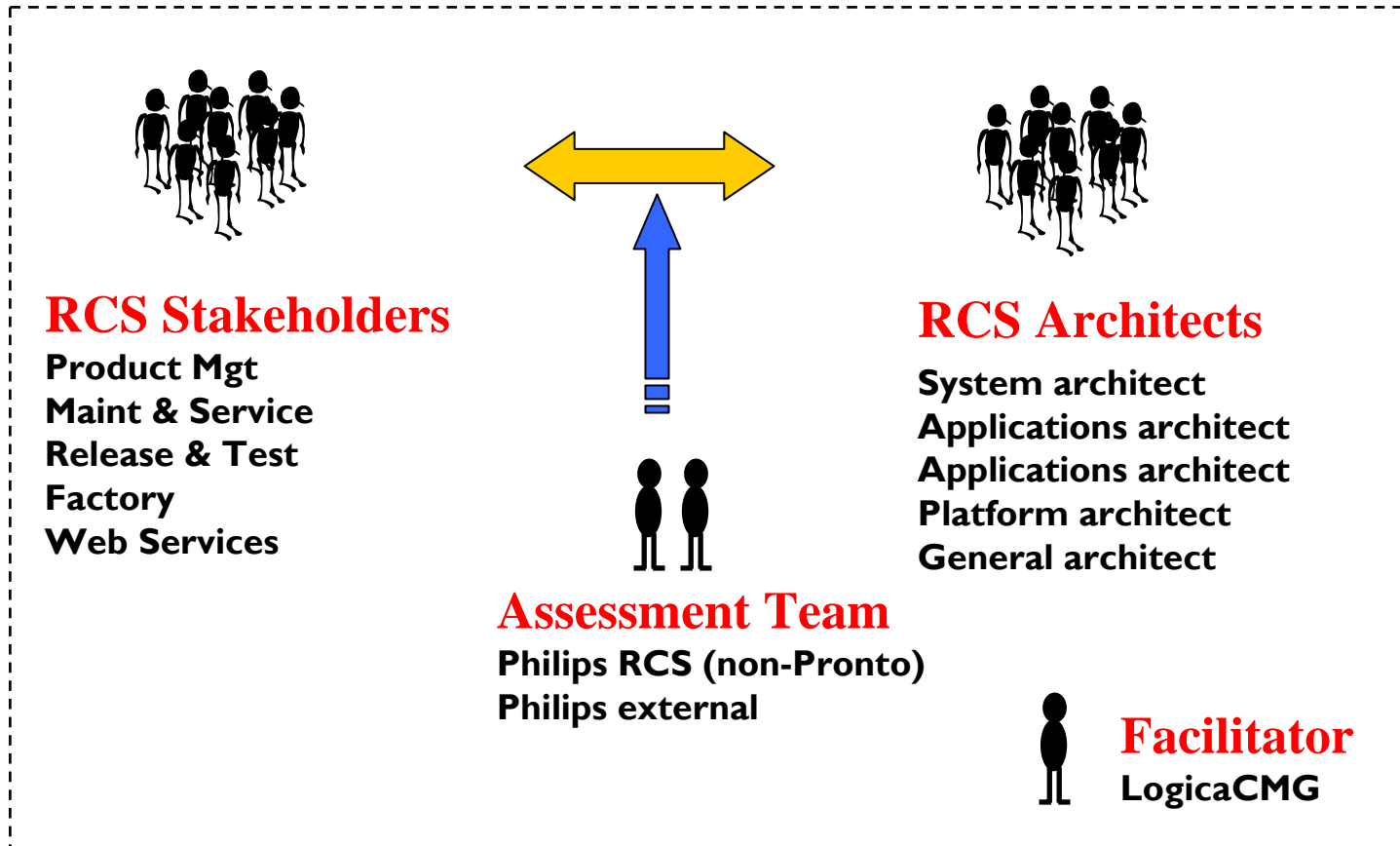


Case : Pronto (high-end Remote Control System)

- ⇒ High-level SW architecture
- ⇒ System, Platform, Application layer

- Identify possible risks in the iP++ SW architecture that may danger successful realization of the project
- Provide recommendations how to effectively address the identified risks
- Customers of the assessment
 - RCS management
 - iP++ stakeholders
 - iP++ SW architects
 - Process management

- High - level SW architecture iPronto++
- SW architecture documentation (informative)
- Aspects to be evaluated
 - Short-term (October release)
 - Critical to success of the project
 - Risk level perceived by architects
 - Architecturally relevant
- Key stakeholders
 - Product Management, Maintenance & Service, SW Release & Testing, Factory, Web Services



Workshop day # 1

1: Kick-off

- Problem statement
- Expectations, benefits, concerns
- Agenda
- Rules of the road

2: Context building

- Global system architecture (bu-view)
- Market perspective, product reqs
- Global system architecture (te-view)

3: Aspect identification

- Relevant aspects + weighing factors
- Short-list of 7 aspects

4: Investigation aspect # 1, 2, 3

- Aspect definition / clarification
- Requirements
- Architecture coverage
- Conclusions, recommendations

Workshop day # 2

4: Investigation aspect # 4

- Investigation aspect # 5
- Investigation aspect # 6
- Investigation aspect # ...
- Investigation aspect # 12

5: Discussion of possible 'hot issues'

6: Summary

- Conclusions
- Recommendations

7: Evaluation

- Expectations, benefits & concerns
- Assessment process

8: Follow-up (reporting phase)

- Approach
- Planning

Selected

- Time performance (1)
- Availability (1)
- Scalability (2)
- Serviceability (3)
- Future proof (3)
- Configurability (4)
- Testability (4)

Not selected

- Support for video (4)
- Internationalization (4)
- Usability (4)
- Interoperability (4)
- Outsourcing (of parts) (4)
- Security (5)
- Reuse / synergy level (5)
- Extend ability (5)
- Product modularity (5)

- **Availability:** Degree to which the system is available to the user when needed
 - Boot time: < 15s
 - Wake-up time: < 1s
 - Response to user action: < 250ms
 - Battery lifetime: 3-4 hrs continuous use
 - Recharge time: 45 minutes

- **Supporting** architectural choices
 - HW peripherals have interrupt/wake up possibility (incl. keypad, touch screen, tilt sensor, ..)
 - After reset, CPU will reboot but kernel will restore RAM; applications will not have to be restarted
- **Obstructing** architectural choices
 - Battery lifetime: deep sleep needed, wake up after reset with Cotulla takes a bit longer than when CPU is on or idle
 - Storage of persistence data will occur before CPU goes to deep sleep; this may introduce a time space in which CPU will not respond to any wake up source

- **Conclusion / Risk**

- Risk medium to high for wake up from deep sleep: assumed it can be done fast enough for user not to notice
- Should be evaluated as soon as power mgt is implemented by MontaVista + hw and drivers are available from Intrinsyc
- Only alternative=CPU always on
=> has severe consequence on battery life

- **Actions to be taken**

- Evaluate wake up time as soon as power mgt sw is available
- To be checked on short term (asking MV/Intrinsyc): can we distinguish a hard reset from a wake up from deep sleep + what is needed to check & assure correct boot behaviour

- Architecture
 - Has a significant impact on product quality
 - Provides an opportunity for early quality prediction
- An architecture assessment
 - Provides an insight into key architectural risks
 - Compensates for a poor architecture process
- Characteristics of the assessment approach
 - Plenary workshop, high level of flexibility
 - Scope limited to the most important subjects
 - Delivers more than architecture and process findings